

Protocolbindingen voor StUF



Versie: 010.0343.02

Status: ~~In Gebruik~~ Concept

Inhoudsopgave

1	Inleiding.....	3
2	Uitwisseling via een bestand.....	5
3	Binding aan SOAP 1.1.....	6
3.1	Het serialiseren van binaire bijlagen.....	6
4	Binding op basis van WSDL, SOAP en http.....	8
4.1	Het gebruik van het <message> element.....	8
4.2	Het gebruik van het <portType> element.....	8
4.3	Het gebruik van het <binding> element.....	10
4.4	Het gebruik van het <service> element.....	11
4.5	Voorschriften voor wsdl-bestanden.....	12
5	Berichtuitwisseling op basis van Digikoppeling profielen.....	13
5.1	Binding aan het Digikoppeling WUS profiel.....	16
5.1.1	WUS 1.1.....	16
5.1.2	WUS 2.0.....	17
5.2	Binding aan het Digikoppeling ebMS profiel	17
5.2.1	ebMS 1.1.....	18
5.2.2	ebMS 2.0.....	20
6	Referenties.....	21

Versiehistorie

Versie	Bestandsnaam	Reden nieuwe versie
03.00 25-06-09	stuf.bindingen.030000.odt	Vastgesteld in de StUF Regiegroep d.d. 24-6-2009
03.00 01-12-09	Stuf.bindingen.030001.odt	ERR0045: Specificatie van het action attribute in de OSB WUS binding verbeterd in de tabel in 5.1 ERR0048: eindtag voor het voorbeeld van een StUF-berichtenset op blz. 4 gelijk gemaakt aan begintag.
03.01 01-03-10	stuf.bindingen.030100.odt	Het definiëren van MTOM als protocol voor het opnemen van base64binary content in de berichten.
- 09-03-10	-	Links die in hoofdstuk 6 'Referenties' nog verwezen naar de EGEM site aangepast.
04.0003.0 2 0810-024- 11	stuf.bindingen.04000030200.odt	<u>Upgrade naar Digikoppeling 2.0 – bestaande implementaties zijn 'upwards-compatible' hiermee.</u> <u>Verwijdering van het WUS-lite protocol omdat de Digikoppeling Gateway niet meer wordt ondersteund.</u>

1 Inleiding

De StUF standaard beschrijft hoe berichten worden gemaakt en wat verwacht mag worden van een StUF-compliant berichtverwerkend systeem. Hierbij is op een hoog functioneel niveau gesproken over berichtcycli. De StUF-standaard standaardiseert de inhoud ("payload") van berichten en laat de keuze van het communicatieprotocol vrij. Dit document beschrijft de binding van StUF-berichtenverkeer aan de volgende protocollen:

1. via bestand;
2. WSDL 1.1 (Web Services Description Language) met SOAP1.1 en http of https als onderliggend transportmechanisme;
3. De protocollen voor ~~de OSBDigikoppeling~~ (~~voorheen OverheidsServiceBusOSB~~).

Een binding van StUF-berichten aan een bestand is van belang voor het efficiënt uitwisselen van grote hoeveelheden berichten. Denk bijvoorbeeld aan het initieel vullen van een database met behulp van StUF-berichten. Bestandsuitwisseling zal gebruikt worden, als interactieve uitwisseling onaantrekkelijk of onmogelijk is, bijvoorbeeld omdat er geen datanetwerk voorhanden is of omdat de capaciteit en performance van het netwerk onvoldoende is of omdat het netwerk te duur is. Hoofdstuk 2 gaat dieper in op het uitwisselen van StUF-berichten via een bestand. Er wordt slechts beschreven hoe berichten in een bestand worden opgenomen. Over de naamgeving en het transport (e-mail, ftp, DVD, USB, etc) van het bestand worden geen uitspraken gedaan.

De binding aan WSDL1.1 en de protocollen van ~~de OSBDigikoppeling~~ maken gebruik van SOAP. Daarom wordt in hoofdstuk 3 gespecificeerd hoe StUF in principe bindt aan SOAP 1.1. Het staat een protocolbinding vrij om op een andere wijze aan SOAP te binden.

Binnen organisaties en, als er geen hoge eisen worden gesteld aan betrouwbaarheid, ook tussen organisaties ligt een keuze voor webservices gebaseerd op WSDL 1.1 met SOAP 1.1 en http als onderliggend transportmechanisme voor de hand vanwege haar laagdrempeligheid en haar brede toepassing in de praktijk. Hoofdstuk 4 gaat dieper op dit protocol in. Hoofdstuk 4 geeft voorschriften en beperkingen voor het specificeren van web services die StUF-berichten synchroon of asynchroon kunnen verwerken.

~~De OSBDigikoppeling~~ heeft ten behoeve van betrouwbare communicatie tussen overheidsorganisatie twee koppelvlak profielen beschreven. Beide profielen bieden voorzieningen voor authenticatie en beveiliging op basis van client en server certificaten. Het gaat om:

1. ~~OSBDigikoppeling~~ WUS ~~1+2.0~~, een koppelvlakprofiel gebaseerd op WUS (de WSDL, UDDI en SOAP stack van W3C-standaarden). ~~OSBDigikoppeling~~ WUS ~~1+2.0~~ is bedoeld voor bevestigingen en biedt geen voorzieningen voor betrouwbaar berichtenverkeer.
2. ~~OSBDigikoppeling~~ ebMS ~~1+2.0~~, een koppelvlakprofiel gebaseerd op ebMS. ~~OSBDigikoppeling~~ ebMS ~~1+2.0~~ is bedoeld voor het doorvoeren van transacties bij de ontvanger en biedt ook voorzieningen voor gegarandeerde aflevering van de berichten.

Digikoppeling 2.0 profielen (WUS en ebMS) zijn een uitbreiding van de Digikoppeling 1.1 profielen. Alle bestaande implementaties van Digikoppeling 1.1 zijn daarom volledig in overeenstemming met Digikoppeling 2.0 (upwards compatible).

~~Om Ook voor kleinere organisaties te helpen bij de implementatie van zijn in de markt tegenwoordig voldoende producten in de markt die als de OSBDigikoppeling-adaptor dienst kunnen doen. Ook zijn er dienstverleners die een volledig beheerde oplossing als dienst aanbieden. De afdeling Markt van Logius levert ondersteuning bij implementatie/selectie. -gateway geïntroduceerd- OSB is de zogenaamde In het verleden bood Logius nog de OSBDigikoppeling Gateway aan als alternatief, maar in de markt zijn tegenwoordig voordeliger oplossingen aanwezig. De bijbehorende protocol-binding is daarom in dit document gehandhaafd ebMSOSB-WUS en OSB-Gateway hoeft niet met certificaten gewerkt te worden. Daarnaast zorgt de gateway voor de benodigde protocoltransformatie van WUS-lite en JMS naar OSB In de communicatie met de . vereenvoudigd koppelvlak naar de interne organisatie wordt vaak ook door andere producten geboden als WUS-lite~~

~~koppelvlakprofiel en een JMS-koppelvlak-profiel~~ ~~zogenamd~~ ~~biedt een~~ Het door deze Gateway geboden ~~Zo'n~~ adapter dient in ieder geval het soap-http protocol zoals beschreven in Hoofdstuk 4 te ondersteunen richting het domein van een gemeente. De adapter zou daarnaast ook andere protocollen zoals JMS kunnen aanbieden indien daar behoefte aan is.

—— Hoofdstuk 5 beschrijft de binding van StUF-berichtenverkeer aan ~~de OSBDigikoppeling~~ koppelvlakprofielen OSBDigikoppeling WUS en OSBDigikoppeling ebMS. Dit document geeft enkele aanvullingen op de OSBDigikoppeling voorschriften voor het definiëren van een OSBDigikoppeling WUS en ebMS koppelvlak voor StUF-berichten.

2 Uitwisseling via een bestand

De via bestand over te dragen StUF -berichten worden in de gewenste verwerkingsvolgorde weggeschreven in een XML-bestand met als root-element <StUF:StUF-berichtenSet> met StUF de namespace prefix voor de namespace URI van de gebruikte StUF-versie. Binnen het <StUF:StUF-berichtenSet> element mogen alleen asynchrone berichten worden opgenomen. Het topelement van een asynchroon bericht dient een namespace prefix te hebben voor de namespace URI van het sectormodel waartoe het bericht behoort. De ontvangst van berichten geleverd in een bestand hoeft niet bevestigd te worden. Omdat een sectormodel eigen namen voor berichten definieert, is het niet mogelijk om in het StUF-schema een nadere definitie van het element <StUF-berichtenSet> te geven. Alle berichten in een StUF-berichtenset dienen gebaseerd te zijn op dezelfde StUF-versie. Het is wel toegestaan om berichten uit verschillende sectormodellen in een StUF-berichtenset op te nemen.

Hieronder staat een voorbeeld van een StUF-berichtenset bestand voor berichten uit het sectormodel BG0310.

```
<StUF:StUF-berichtenSet xmlns:StUF="http://www.egem.nl/StUF/StUF0301"
xmlns:BG="http://www.egem.nl/StUF/sector/BG/0310">
  <BG:npsLk01>
    ...
  </BG:npsLk01>
  ...
  <BG:aoaLk01>
    ...
  </BG:aoaLk01>
  ...
</StUF:StUF-berichtenSet>
```

In een bestand kunnen grote hoeveelheden StUF-berichten worden opgeslagen. StUF geeft geen voorschriften voor het splitsen van een groot bestand in deelbestanden. Technieken als TAR, ZIP en RAR bieden functionaliteit om een bestand te splitsen. De ontvanger zal de bestanden zonodig weer samenvoegen en aan de geadresseerde applicatie aanbieden. StUF geeft ook geen voorschriften voor het transport van een StUF-berichtenset. We zullen in dit document verder geen aandacht besteden aan deze vorm van uitwisseling.

3 e-bD Binding aan SOAP 1.1

SOAP 1.1 (Simple Object Access Protocol) wordt gebruikt voor het verpakken van een StUF-bericht ten behoeve van het transport over internet. Het StUF-bericht wordt opgenomen in het <SOAP:body> element in het <SOAP:envelope> element. Het <SOAP:body> element kan binnen het <SOAP:envelope> element worden voorafgegaan door nul of meer <SOAP:header> elementen. Een <SOAP:header> element is bedoeld voor de sturing van het transport en hoog niveau foutafhandeling. Het <SOAP:body> element is bedoeld voor de payload. Eventuele voorschriften voor de <SOAP:header> elementen worden gegeven bij de beschrijving van een specifieke protocolbinding.

SOAP voorziet via het attribute `encodingStyle` in de mogelijkheid verschillende vormen van serialisatie te ondersteunen. Conform WS-I BP 1.1 [WSIBP] verbiedt StUF het gebruik van het attribute `encodingStyle` binnen een wsdl voor StUF-berichten.

Ten behoeve van foutafhandeling kent SOAP het <SOAP:fault> element. De elementen van het <SOAP:fault> element worden gevuld op basis van de elementen binnen het StUF-foutbericht:

- Het element <SOAP:faultcode> wordt gevuld met de namespace qualifier voor de namespace “<http://schemas.xmlsoap.org/soap/envelope/>” gevolgd door een ':' en de waarde van <StUF:plek> binnen het foutbericht, bijvoorbeeld “SOAP-ENV:Client”.
- Het element <SOAP:faultstring> wordt gevuld met de waarde van <StUF:omschrijving> binnen het foutbericht.
- Het element <SOAP:details> wordt gevuld met het StUF-foutbericht.
- Het element <SOAP:faultactor> wordt niet opgenomen in het <SOAP:fault> element. ~~Het is de verantwoordelijkheid van het systeem dat een verzoek verwerkt om het <SOAP:fault> element correct te vullen.~~
-

3.1 Het serialiseren van binaire bijlagen

Binaire bijlagen kunnen in een XML-bericht worden opgenomen door middel van elementen van het type base64binary uit de namespace “<http://www.w3.org/2001/XMLSchema>”. Aan het versturen van bijlagen binnen het XML-bericht kleven enkele nadelen en om deze te ondervangen is MTOM (SOAP Message Transmission Optimization) ontwikkeld in combinatie met xop (XML-binary Optimized Packaging). De volgende links bevatten relevante informatie met betrekking tot MTOM/xop:

- <http://www.w3.org/TR/soap12-mtom/> (specificatie van MTOM in combinatie met SOAP1.2)
- <http://www.w3.org/TR/xop10/> (specificatie van xop gebruikt door MTOM)
- <http://www.w3.org/Submission/soap11mtom10> (specificatie van het gebruik van MTOM in combinatie met SOAP1.1)
- <http://www.w3.org/TR/xml-media-types/> (specificatie van het specificeren van MIME mediatype in berichten en schema's)

Deze protocolbinding definieert in de StUF-namespaces “<http://www.egem.nl/StUF/StUF0204>” en “<http://www.egem.nl/StUF/StUF0301>” de complexTypes BinaireInhoud-basis, BinaireInhoud en BinaireInhoud-vraag, een attribute bestandsnaam en een simpleType Bestandsnaam:

```
<complexType name="BinaireInhoud">
  <simpleContent>
    <restriction base="StUF:BinaireInhoud-basis">
      <attribute ref="StUF:bestandsnaam" use="required"/>
    </restriction>
  </simpleContent>
</complexType>
<complexType name="BinaireInhoud-basis">
  <simpleContent>
    <extension base="xmime:base64Binary">
```

```
<attribute ref="StUF:bestandsnaam"/>
</extension>
</simpleContent>
</complexType>
<complexType name="BinaireInhoud-vraag">
  <simpleContent>
    <restriction base="StUF:BinaireInhoud-basis">
      <attribute ref="xmime:contentType" use="prohibited"/>
      <attribute ref="StUF:bestandsnaam" use="prohibited"/>
    </restriction>
  </simpleContent>
</complexType>
<attribute name="bestandsnaam" type="StUF:Bestandsnaam"/>
<simpleType name="Bestandsnaam">
  <restriction base="string">
    <maxLength value="255"/>
  </restriction>
</simpleType>
```

Hierbij is `xmime` een prefix voor de namespace "<http://www.w3.org/2005/05/xmlmime>". Er is gekozen voor `xmime:base64Binary` als base waaraan het attribute `StUF:bestandsnaam` wordt toegevoegd, omdat dan in het optionele attribute `xmime:contentType` het MIME contenttype kan worden meegegeven. Er is gekozen om de bestandsnaam van de binaire bijlage verplicht op te nemen, omdat de bestandsnaam informatie geeft over de programmatuur waarmee een binaire bijlage kan worden verwerkt. Deze informatie kan lang niet altijd in het attribute `xmime:contentType` worden gecodeerd, omdat niet alle bestandsformaten worden ondersteund (Microsoft Word bijvoorbeeld niet).

Er zijn drie complexTypes voor `BinaireInhoud`, omdat je niet wilt dat in de selectie- en scope-elementen in een vraagbericht attributes worden opgenomen. Binnen het basis-complexType voor een `StUF`-entiteittype dient `BinaireInhoud-basis` gebruikt te worden en binnen alle complexTypes die in een concreet bericht kunnen voorkomen het complexType `BinaireInhoud`, met uitzondering van de selectie- en scope-elementen in een vraagbericht, waar het complexType `BinaireInhoud-vraag` gebruikt dient te worden.

De bovenstaande complexTypes, attribute en simpleType worden gedefinieerd in een bestand `stuf0204mtom.xsd` en `stuf0301mtom.xsd` met als targetnamespace "<http://www.egem.nl/StUF/StUF0204>" respectievelijk "<http://www.egem.nl/StUF/StUF0301>". Deze schema's definiëren een uitbreiding op de alleen voor een bugfix wijzigbare `stuf0204.xsd` en `stuf0301.xsd` en bevatten via een include deze schema's.

Deze protocolbinding schrijft voor:

1. De hier gegeven voorschriften voor het serialiseren van binaire bijlagen dienen gevolgd te worden, als voor de namespace `StUF` als schema '`stuf0204mtom.xsd`' of '`stuf0301mtom.xsd`' wordt gebruikt.
2. Een element met content van het type `base64binary` mag uitsluitend in een `StUF`-bericht worden opgenomen door middel van het complexType `BinaireInhoud` uit de `StUF`-namespace.
3. Als in een bericht content voorkomt van het type `base64binary`, dan dient dit bericht geserialiseerd te worden conform de MTOM specificatie voor SOAP1.1 ongeacht de lengte van de `base64binary` content.
4. Partijen die een sectormodel met deze protocolbinding ondersteunen, dienen MTOM te ondersteunen bij zowel het verzenden als het ontvangen van berichten.
5. Voor wat betreft een maximum grootte van een binaire bijlage, conformeert deze protocolbinding zich aan de voorschriften van de OSB of Digikoppeling.

4 StUF-bBinding op basis van WSDL, SOAP en http

In deze binding wordt de uitwisseling van StUF-berichten gedefinieerd met behulp van WSDL (versie 1.1, zie [WSDL]). In het vervolg zal deze protocolbinding worden aangeduid als de StUF http/SOAP binding. Deze binding voldoet aan het WS-I Basic Profile 1.1 [WSIBP]. Deze binding is bedoeld voor het definiëren van eenvoudige web services. Er wordt niet ingegaan op eisen op het gebied van betrouwbaarheid, beveiliging en authenticatie. Het staat gebruikers van deze binding vrij om hiervoor aanvullende eisen te stellen, maar het is wel vereist dat bindingen die WSDL gebruiken voortbouwen op dit hoofdstuk. De hier gegeven specificaties mogen aangevuld worden, maar niet afgezwakt.

Functioneel kent StUF de interactiepatronen voor een asynchrone notificatie (er wordt geen [functionele](#) respons terugverwacht) en voor verzoek/respons in de smaken synchroon en asynchroon. De StUF-standaard schrijft in verband met Quality of Service voor, dat op een asynchrone notificatie en op een asynchroon verzoek of een asynchrone respons in geval van fouten wordt gereageerd met een Fo03-foutbericht. Als er geen problemen zijn, dan dient de serviceverlener [op de een-of-andere manier aan-aan](#) de ontvanger te laten weten dat het bericht in goede orde ontvangen is. De StUF-standaard heeft voor dit doel het Bv03-bevestigingsbericht (de ontvanger heeft de stuurgegevens gecheckt en denkt het bericht te kunnen verwerken en heeft het bericht veilig opgeslagen) en het Bv04-bevestigingsbericht (de ontvanger heeft het bericht in goede orde ontvangen en veilig opgeslagen, maar nog helemaal niet gecheckt) gedefinieerd. Op het niveau van de servicedefinitie in het wsdl-bestand is er dus ook voor asynchrone berichten een respons gedefinieerd.

In deze binding wordt http of https als transportprotocol gebruikt. Binnen http en https wordt op protocolniveau op elk verzoek gereageerd met een respons. Bij gebruik van http of https kan daarom voor wat betreft time outs op de connectie worden aangesloten op de connection time out foutafhandeling van http. Als op http niveau een connection time out fout wordt gegeven, dan dient de zender ervan uit te gaan, dat het verzonden bericht niet is aangekomen en het op een later tijdstip opnieuw te versturen.

De StUF http/SOAP binding definieert geen additionele foutafhandeling boven op de standaard foutafhandeling rond time-outs voor http. Er hoeft bijvoorbeeld geen speciaal foutbericht gestuurd te worden, indien een synchrone vraag niet binnen de time-out tijd voor http-communicatie beantwoord kan worden.

In het vervolg van dit hoofdstuk zal worden ingegaan op het gebruik van de WSDL-structuren `<message>`, `<portType>`, `<binding>` en `<service>`. Het staat gebruikers van de StUF http/SOAP binding vrij om in de wsdl of daarbuiten extra specificaties op te nemen voor het gebruik van `<SOAP:header>` en `<SOAP:headerfault>` elementen. In het hoofdstuk over de binding aan [dehet OSB-Digikoppeling](#) WUS profiel wordt bijvoorbeeld nader gespecificeerd hoe een `<SOAP:header>` element met WS-Addressing gegevens gevuld dient te worden.

4.1 Het gebruik van het `<message>` element

Elk StUF-bericht dat als verzoek- of responsbericht wordt gebruikt, dient als `<part>` binnen een `<message>` element gedefinieerd te worden. De name voor het `<part>` is "body". Er moet via het attribute `element` verwezen worden naar het element voor het bericht in het sectormodel, omdat anders niet voldaan wordt aan het WS-I Basic Profile 1.1. Het attribute `name` binnen `<message>` heeft als waarde de naam van het element waarnaar verwezen wordt.

De definities van de `<message>` elementen voor de in StUF-standaard gedefinieerde bevestigingsberichten en foutberichten zijn opgenomen in het bestand `stuf0301.types.wsdl` met als namespace "http://www.egem.nl/StUF/StUF0301". In dit bestand zijn ook opgenomen de door de StUF-standaard gedefinieerde interactiepatronen. Dit bestand kan worden geïmporteerd in de wsdl-bestanden behorend bij een sectormodel.

4.2 Het gebruik van het `<portType>` element

Binnen een wsdl mag conform de voorschriften van [de OSB-Digikoppeling](#) slechts één `<portType>` element worden opgenomen. Er dient dus per onderkend `<portType>` een wsdl gemaakt te worden.

Binnen het `<portType>` element worden in de vorm van `<operation>` elementen de ondersteunde interactiepatronen functioneel gedefinieerd. Om ervoor te zorgen dat eenvoudig met verschillende systemen die StUF implementeren gecommuniceerd kan worden, worden voor de StUF onderkende interactiepatronen `<portType>` elementen voorgeschreven. Een systeem is verplicht zo'n voorgeschreven `<portType>` element te ondersteunen, als het zo'n interactiepatroon ondersteunt. Het staat een systeem vrij om voor een binnen een voorgeschreven `<portType>` element ondersteund interactiepatroon in een andere wsdl ook `<portType>` elementen te definiëren die dit interactiepatroon ook ondersteunen.

Een `<operation>` element voor een StUF-bericht binnen een voorgeschreven `<portType>` element bevat als elementen `<input>`, `<output>` en nul of één keer `<fault>`. Het attribute `name` van `<operation>` en het attribute `message` van `<input>` worden gevuld met het attribute `name` van het `<message>` element voor het inkomende bericht, binnen `<input>` uiteraard gekwalificeerd met de namespace kwalifier voor de target namespace van de wsdl. Voor de voorgeschreven portTypes wordt hieronder beschreven hoe het attribute `name` wordt gevuld binnen de elementen `<output>` en `<fault>`.

De voorgeschreven `<portType>` elementen zijn:

- **OntvangAsynchroon**
Het element `<portType>` heeft als waarde voor het name attribute 'OntvangAsynchroon'. Binnen dit `<portType>` worden `<operation>` elementen gedefinieerd voor alle asynchrone berichten, die het systeem kan ontvangen. Het element `<output>` heeft als inhoud: `<output message="StUF:Bv03"/>` en het element `<fault>` heeft als inhoud `<fault name="fout" message="StUF:Fo03"/>` met StUF de namespace kwalifier voor de namespace "http://www.egem.nl/StUF/StUF0301". Als een systeem naar andere systemen asynchrone verzoek/respons berichten stuurt, mag niet vergeten worden `<operation>` elementen te definiëren voor de Bv01- en Fo01-berichten die als respons op een asynchroon verzoek gestuurd kunnen worden.
- **VerwerkSynchroneKennisgeving**
Het element `<portType>` heeft als waarde voor het name attribute 'VerwerkSynchroneKennisgeving'. Binnen dit `<portType>` worden `<operation>` elementen gedefinieerd voor alle synchrone kennisgeving- en synchronisatieberichten die het systeem ondersteunt. Het element `<output>` heeft als inhoud: `<output message="StUF:Bv02"/>` en het element `<fault>` heeft als inhoud `<fault name="fout" message="StUF:Fo02"/>` met StUF de namespace kwalifier voor de namespace "http://www.egem.nl/StUF/StUF0301".
- **VerstrekSynchronisatieBericht**
Het element `<portType>` heeft als waarde voor het name attribute 'VerstrekSynchronisatieBericht'. Binnen dit `<portType>` worden `<operation>` elementen gedefinieerd met het attribute `message` in het element `<input>` gevuld met "tns:Sa04" of "tns:Sh04" met tns de namespace kwalifier voor de target namespace van de wsdl. In het element `<output>` wordt het attribute `message` gevuld met "tns:Sa02" respectievelijk "tns:Sh02". Het element `<fault>` heeft als inhoud `<fault name="fout" message="StUF:Fo02"/>` met StUF de namespace kwalifier voor de namespace "http://www.egem.nl/StUF/StUF0301".
- **BeantwoordVraag**
Het element `<portType>` heeft als waarde voor het name attribute 'BeantwoordVraag'. Binnen dit `<portType>` worden `<operation>` elementen gedefinieerd voor alle synchrone vraagberichten die het systeem kan beantwoorden. In het element `<output>` wordt het attribute `message` gevuld met de elementnaam van het antwoordbericht gekwalificeerd met de targetnamespace voor de wsdl. Het element `<fault>` heeft als inhoud `<fault name="fout" message="StUF:Fo02"/>` met StUF de namespace kwalifier voor de namespace "http://www.egem.nl/StUF/StUF0301".
- **VerwerkTriggerbericht**
Het portType `<portType name="VerwerkTriggerbericht">` voor het verwerken van het triggerbericht is gedefinieerd in de wsdl stuf0301.types.wsdl. Het element `<input>` verwijst naar het Tr01-bericht, het element `<output>` naar het Bv02-bericht en het element `<fault>` naar het Fo02-bericht.

- VerwerkSynchroonVrijBericht

Het element `<portType>` heeft als waarde voor het name attribute 'VerwerkSynchroonVrijBericht'. Binnen dit `<portType>` worden `<operation>` elementen gedefinieerd voor alle synchrone vrije verzoekberichten die het systeem ondersteunt. In het element `<output>` wordt het attribute message gevuld met de elementnaam van het vrije responsbericht gekwalificeerd met de namespace qualifier voor de target namespace van de wsdl. Het element `<fault>` heeft als inhoud `<fault name="fout" message="StUF:F002"/>` met StUF de namespace qualifier voor de namespace "http://www.egem.nl/StUF/StUF0301".

Een intermediaire node dient zich ook te houden aan de bovenstaande voorschriften. Een intermediaire node mag uitsluitend portTypes en operations aanbieden, die de intermediaire node voor verwerking kan doorsturen naar een andere intermediaire node of een StUF-endpoint. Voor het `<portType>` OntvangAsynchroon gelden op een intermediaire node licht afwijkende voorschriften. Binnen het `<portType>` OntvangAsynchroon worden `<operation>` elementen gedefinieerd voor alle asynchrone berichten, die de intermediaire node kan doorsturen. Het element `<output>` binnen een operation op een intermediaire node heeft als inhoud `<output message="StUF:Bv04"/>`. Het voor een StUF endpoint voorgeschreven element `<fault>` is voor een intermediaire node niet verplicht. Een ander `<fault>` element mag niet voorkomen.

4.3 Het gebruik van het `<binding>` element

Het `<binding>` element specificeert met welk protocol en hoe exact binnen dat protocol een interactiepatroon is geïmplementeerd, dat binnen een `<portType>` in een `<operation>` is gedefinieerd. Een `<binding>` wordt altijd gemaakt voor één `<portType>`. StUF schrijft voor dat er voor een `<portType>` element precies één `<binding>` element is, dat SOAP en http als protocol voorschrijft. De `<binding>` element heeft als waarde voor het attribute name 'SOAP' geconcateneerd met de naam voor het `<portType>`. Bijvoorbeeld voor het `<portType name="BeantwoordVraag">` wordt het `<binding>` element dan

```
<binding name="SOAPBeantwoordVraag" type="tns:BeantwoordVraag">
```

met tns de namespace qualifier van de targetnamespace voor de wsdl. Voor zelf gedefinieerde `<portType>` elementen mogen meerdere `<binding>` elementen worden gebruikt.

De binding aan SOAP en http wordt gespecificeerd door als kind binnen het `<binding>` element op te nemen

```
<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
```

met soap de namespace qualifier voor de namespace "http://schemas.xmlsoap.org/wsdl/soap". WS-I BP 1.1 schrijft document style voor en verbiedt rpc style.

Daarnaast wordt voor de te binden `<operation>` elementen uit het `<portType>` een `<operation name="nmtoken">` element binnen het `<binding>` element opgenomen met nmtoken de naam van het `<operation>` element binnen het `<portType>`. Binnen dit `<operation>` element wordt gespecificeerd hoe de berichten verzonden worden. Het eerste element binnen een `<operation>` element in een `<binding>` element is

```
<soap:operation soapAction="xxx"/>
```

Het attribute soapAction wordt gevuld met de namespaceUri van het element gespecificeerd in het body part van de input message van de operation gevolgd door "/" en door "nmtoken" uit het element `<operation name="nmtoken">` binnen het `<binding>` element. Het attribute soapAction wordt dus gevuld met "http://www.egem.nl/StUF/sector/bg/0310/npsLk01" voor een `<operation`

name="npsLk01"> element in het <portType> element voor een adreskennisgeving uit het sectormodel bg0310.¹

In de elementen <input>, <output> en <fault> wordt aangegeven hoe de <SOAP:header> en <SOAP:body> elementen in de SOAP-envelope gevuld. Het <output> element definieert de SOAP-envelope voor de normale respons op het inkomend bericht en het <fault> element definieert het <SOAP:detail> element in geval van een foutsituatie. De inhoud van het <SOAP:body> element voor het verzoek en de gewone respons wordt gespecificeerd in het element

```
<SOAP:body use="literal"/>
```

binnen <input> en <output>.

Het parts attribute binnen <SOAP:body> is niet nodig, omdat de <message> behorend bij de <operation> slechts één <part> bevat. Het verplichte use attribute in <SOAP:body> geeft met de waarde literal samen met het ontbreken van het encodingStyle attribute aan dat er geen additionele encoding-regels zijn.

De inhoud van <fault> elementen binnen een <operation> specificeert hoe het <SOAP:detail> element binnen het <SOAP:detail> element moet worden gevuld in geval van een fout. Binnen <operation> elementen in een <portType> voor StUF komt altijd slechts één <fault> element voor met name="fout". Een <operation> element in een <binding> bevat daarom gegeven de specificatie in hoofdstuk 3 altijd precies één <fault> element met als inhoud

```
<fault name="fout">  
  <SOAP:detail name="fout" use="literal"/>  
</fault>
```

Een voorbeeld van een <operation> element voor de ontvangst van asynchrone kennisgeving voor een natuurlijk persoon staat hieronder:

```
<operation name="npsLk01">  
  <soap:operation soapAction="http://www.egem.nl/StUF/sector/bg/0310/npsLk01"/>  
  <input>  
    <soap:body use="literal"/>  
  </input>  
  <output>  
    <soap:body use="literal"/>  
  </output>  
  <fault name="fout">  
    <soap:detail name="fout" use="literal"/>  
  </fault>  
</operation>
```

Het enige dat hierin varieert is de waarde van het name attribute voor de <operation> en de waarde van het soapAction attribute.

4.4 Het gebruik van het <service> element

Binnen het <service> element kunnen in de vorm van <port> elementen één of meer gerelateerde 'communication endpoints' voor een <binding> worden gedefinieerd. Omdat StUF voorschrijft, dat een wsdl slechts één <portType> en één <binding> element mag bevatten, zal een <service> element altijd slechts één <port> element bevatten.

¹ Er is niet voor gekozen om de default vulling van was:Action uit WS-Addressing Metadata standaard te volgen, omdat de hier gegeven definitie simpeler is en ook garandeert dat de SOAPAction een absolute IRI is.

Het attribute name op het <service> element en het attribute name op het <port> element krijgen dezelfde waarde als het name attribute van het <portType>. Het attribute binding op het <port> element krijgt als waarde de namespace qualifier voor de target namespace van de wsdl gevolgd door de waarde van het name attribute van het <binding> element. In het element <soap:address> specificeert het attribute location waar de webservice te vinden is. De waarde van het location attribute dient te eindigen met een '/' gevolgd door de waarde van het name attribute van het <portType>, <service> en <port> element. Hiervoor dient een URI te staan die in alle door StUF voorgeschreven portType-wsdl's voor één sectormodel identiek is. Verschillende sectormodellen mogen deze URI delen.

Een voorbeeld van het <service> element voor het portType OntvangAsynchroon staat hieronder.

```
<service name="BeantwoordVraag">
  <port name="BeantwoordVraag" binding="tns:SOAPBeantwoordVraag ">
    <soap:address location="http://example.com/BeantwoordVraag"/>
  </port>
</service>
```

4.5 Voorschriften voor de inhoud en naamgeving van devoor wsdl-bestanden

Voor elk van de in paragraaf 4.2 gedefinieerde <portType> elementen, dient een aparte wsdl gemaakt te worden. In deze wsdl-file worden ook opgenomen de <message> elementen met de definitie van de binnen het <portType> element gebruikte berichtelementen en het <binding> element. De berichtelementen worden niet opgenomen in de wsdl-file maar in een xsd binnen het sectormodel. Deze xsd wordt in de wsdl geïmporteerd conform WS-I BP1.1 voorschriften hiervoor.

De naam van het wsdl-bestand dient te eindigen met de code voor het sectormodel inclusief viercijferig versienummer gevolgd door '.xxx.wsdl' met xxx de de naar kleine letters omgezette waarde van het name attribute voor het <portType>. Er staat eindigen, omdat verschillende systemen binnen een organisatie verschillende wsdl's kunnen gebruiken.

Een en hetzelfde <portType> element kan worden gebruikt binnen verschillende servicedefinities. In dat geval is het handig om <message>, <portType> en <binding> in een afzonderlijke wsdl te definiëren en te importeren in de service-wsdl. In de praktijk is de behoefte aan importeren niet zo groot, omdat verschillende implementaties lang niet altijd dezelfde verzameling operaties ondersteunen. De naam van een wsdl met de abstracte servicedefinitie eindigt met sectormodelversie.xxx.abstract.wsdl. De wsdl voor het beantwoorden van synchrone vraagberichten uit het sectormodel bg0310 eindigt bijvoorbeeld op bg0310.beantwoordvraag.wsdl. De eventuele wsdl met de abstracte servicedefinitie (excl. het <service> element) eindigt dan op bg0310.beantwoordvraag.abstract.wsdl. In bestandsnamen worden uitsluitend kleine letters gebruikt, omdat sommige operating systemen werken met hoofd en kleine letter gevoelige bestandsnamen.

5 Berichtuitwisseling op basis van de OSBDigikoppeling profielen

De OverheidsServiceBus of OSBDigikoppeling is een verzameling standaarden en afspraken die de berichtuitwisseling tussen overheidsorganisaties beschrijven. Zulke standaarden en afspraken worden ook wel koppelvlakstandaarden genoemd. Dit hoofdstuk beschrijft de binding van StUF-berichten aan de OSBDigikoppeling en is gebaseerd op versie 1.1 van de OSBDigikoppeling koppelvlak standaarden die zijn te vinden op www.logius.nl www.overheidsservicebus.nl.

De OSBDigikoppeling onderkent twee vormen van berichtuitwisseling:

- Het raadplegen van gegevens over een synchrone http request/response verbinding zonder betrouwbare en gegarandeerde overdracht. De koppelvlakstandaard hiervoor is OSBDigikoppeling WUS.
- Het gegarandeerd overdragen van een bericht, waarbij een eventuele functionele respons asynchroon wordt gegeven in een op een later tijdstip verzonden en ook weer gegarandeerd over te dragen bericht. De koppelvlakstandaard hiervoor is OSBDigikoppeling ebMS. OSBDigikoppeling ebMS wordt veelal gebruikt voor het door de zender van een bericht laten doorvoeren van een transactie bij de ontvanger van een bericht.

StUF kent een groot aantal functioneel verschillende interactiepatronen of berichttypen, die op hoofdlijnen te classificeren zijn in de volgende categorieën:

- Lees-berichten: de ontvanger retourneert informatie op basis van een verzoek maar voert geen wijzigingen door in zijn systeem.
- Schrijf-berichten: de ontvanger voert verplicht wijzigingen door in het eigen systeem op basis van een verzoek en retourneert al (synchroon) dan niet (asynchroon) informatie naar de zender. StUF kennisgevingen met indicatorOvername 'V' en StUF synchronisatieberichten worden gekenmerkt als 'schrijf berichten'.
- Melding-berichten: dit zijn berichten waarbij de ontvanger desgewenst wijzigingen in het eigen systeem doorvoert en waarop geen response verwacht wordt (bijvoorbeeld kennisgevingen met indicatorOvername 'I').

Daarnaast kent StUF synchrone en asynchrone berichtuitwisseling. Synchroon is gericht op verzoek-respons waarbij de respons wordt gegeven over dezelfde verbinding als waarop het verzoek is binnengekomen. De respons dient gegeven te worden binnen een time-out tijd karakteristiek voor de verbinding. Asynchroon is gericht op een melding of verzoek-respons waarbij de respons wordt gegeven over een andere verbinding dan waarover het verzoek is ontvangen. Een asynchrone respons wordt vaak pas na verloop van enige tijd gegeven. De zender van het bericht wacht er niet op en gaat verder met zijn eigen processen. Omdat de respons over een andere verbinding wordt gegeven dan het verzoek is bij asynchrone berichten de correlatie tussen het verzoek- en responsbericht belangrijk.

StUF beveelt ook bij asynchrone berichtuitwisseling een respons aan. De zender kan aan de hand van deze respons bepalen of het bericht bij de ontvanger is aangekomen en door hem verwerkt kan worden.

Bij de binding van StUF-berichten aan OSBDigikoppeling-profielen is allereerst van belang of een bericht bindt aan het OSBDigikoppeling WUS of het OSBDigikoppeling ebMS profiel. De onderstaande matrix geeft de binding van StUF-berichten aan de OSBDigikoppeling WUS en ebMS profielen op basis van de bovenstaande classificatie van StUF berichten.

	Synchroon	Asynchroon
StUF Lees berichten	OSBDigikoppeling WUS	OSBDigikoppeling ebMS
StUF Schrijf berichten	-	OSBDigikoppeling ebMS
StUF Melding berichten	-	OSBDigikoppeling ebMS

Tabel 1: Uitgangspunten voor de binding van StUF-berichten aan OSB profielen

Asynchrone berichten binden sowieso aan het [OSBDigikoppeling](#) ebMS profiel, omdat voor asynchrone berichten betrouwbare overdracht essentieel is. Synchrone StUF Schrijf- en Meldingberichten worden niet gebonden aan een [OSBDigikoppeling](#) profiel. Volgens [de OSBDigikoppeling](#) mogen dergelijke berichten niet uitgewisseld worden via [OSBDigikoppeling](#) WUS. De StUF standaard beveelt aan om synchrone berichten direct te verzenden naar de end node die het bericht afhandelt. Bij verzending via ebMS zitten de ebMS adapters hier als intermediairs tussen in. De synchrone StUF Lees-berichten binden aan het [OSBDigikoppeling](#) WUS profiel.

De onderstaande tabel geeft voor elke StUF berichtcode de mapping naar een [OSBDigikoppeling](#) profiel aan. Voor verzoekberichten met een respons wordt eerst de berichtcode van het verzoek gegeven gevolgd door een '/' en de berichtcode van de respons. Een foutbericht als respons wordt niet vermeld. Als er bij een bericht geen respons is gedefinieerd, dan verwacht de verzender geen functionele respons. Bovenin de tabel zijn eerst de door de StUF-standaard gedefinieerd responsberichten opgenomen: de bevestigings- (Bv0n) en foutberichten (Fo0n).

StUF Berichtcode	OSBDigikoppeling Profiel	Toelichting
Bv01 Fo01	OSBDigikoppeling ebMS	een bevestigings- (Bv01) of foutbericht (Fo01) als functionele asynchrone respons.
Bv02 Fo02	OSBDigikoppeling WUS	een bevestigings- (Bv02) of foutbericht (Fo02) als functionele synchrone respons op een synchroon bericht. De Bv02- en Fo02-berichten komen alleen voor als responsbericht. De Fo02-berichten worden bij de andere berichten niet als mogelijke respons genoemd in deze tabel.
Bv03 Fo03	-	<p>een bevestigings- (Bv03) of foutbericht (Fo03) als technische synchrone respons bij verzending van een asynchroon bericht, waarbij de ontvanger de verwerkbaarheid van het bericht gecheckt heeft.</p> <p>Het Bv03-bericht vervalt binnen OSBDigikoppeling ebMS, omdat asynchroon bevestigen dat een ontvangen bericht verwerkbaar is weinig zin heeft en omdat een deel van de door StUF gespecificeerde controles sowieso plaats vindt op ebMS niveau, mits de CPA goed is ingericht. De ontvanger kan net zo goed wachten op de functionele respons op het bericht. Het Fo03-bericht is ook binnen OSBDigikoppeling ebMS een geldige functionele respons en blijft dus bestaan. Bij de binding aan het ebMS protocol is het de verantwoordelijkheid van de ontvanger om het Fo03-bericht aan te bieden aan de ebMS adapter. Het Fo03-bericht zal als <u>asynchroon</u> bericht worden aangeboden aan de verzender van het verzoek.</p>
Bv04	-	<p>een bevestigingsbericht als technische synchrone respons bij verzending van een asynchroon bericht, waarbij de ontvanger slechts garandeert dat het bericht ontvangen is. Er wordt niet gecheckt op de verwerkbaarheid door de ontvanger.</p> <p>Het Bv04-bericht vervalt. De ebMS adapter dient op een of andere manier aan de aanbieder van een bericht te garanderen, dat het bericht zal worden afgeleverd.</p>
Di01/Du01	OSBDigikoppeling ebMS	een asynchroon verzoek (Di01)/respons (Du01) vrij bericht
Di02/Du02	OSBDigikoppeling ebMS	een synchroon verzoek (Di02)/respons (Du02) vrij bericht.

StUF Berichtcode	OSBDigikoppeling Profiel	Toelichting
	of OSBDigikoppeling WUS	Het OSBDigikoppeling berichttype is afhankelijk van de classificatie van het StUF-bericht als leesbericht. Een leesbericht wordt gebonden aan het OSBDigikoppeling WUS profiel. Synchrone schrijf- en meldingberichten kunnen niet gebonden worden aan een OSBDigikoppeling profiel. In plaats hiervan dient de asynchrone variant gebruikt te worden.
LvN/LaN	OSBDigikoppeling WUS (N oneven) OSBDigikoppeling ebMS (N even)	synchroon (N oneven) of asynchroon (N even) vraag/antwoord berichten.
Lk0N	OSBDigikoppeling ebMS	N oneven: een asynchroon kennisgevingbericht
Lk0N/Bv02	-	N even: Deze berichten kunnen als synchrone schrijfberichten niet over de OSBDigikoppeling gebruikt worden. In plaats hiervan dient de asynchrone variant gebruikt te worden.
Sa01	OSBDigikoppeling ebMS	een asynchroon synchronisatiebericht voor de actuele gegevens
Sa02/Bv02	-	Deze berichten kunnen niet over de OSBDigikoppeling gebruikt worden. In plaats hiervan dient de asynchrone variant gebruikt te worden.
Sa03/Sa01	OSBDigikoppeling ebMS	een asynchroon bericht dat vraagt om een asynchroon synchronisatiebericht voor de actuele gegevens
Sa04/Sa02	OSBDigikoppeling WUS	een synchroon bericht dat vraagt om een synchroon synchronisatiebericht voor de actuele gegevens
Sh01	OSBDigikoppeling ebMS	een asynchroon synchronisatiebericht voor de actuele en de historische gegevens
Sh02/Bv02	-	Deze berichten kunnen niet over de OSBDigikoppeling gebruikt worden. In plaats hiervan dient de asynchrone variant gebruikt te worden.
Sh03/Sh01	OSBDigikoppeling ebMS	een asynchroon bericht dat vraagt om een asynchroon synchronisatiebericht voor de actuele en historische gegevens
Sh04/Sh02	OSBDigikoppeling WUS	een synchroon bericht dat vraagt om een synchroon synchronisatiebericht voor de actuele en historische gegevens
Tr01/Bv02	-	een triggerbericht Het Tr01-bericht kan vervallen, omdat de logistieke laag verantwoordelijk is voor de buffering van asynchrone berichten en de aflevering van berichten uit de buffer.

Tabel 2: De binding van StUF-berichten aan OSBDigikoppeling profielen

Overheidsorganisaties die via de OSBDigikoppeling met elkaar willen communiceren dienen te beschikken over een zogenaamd OverheidsIdentificatieNummer of OIN. Voor communicatie tussen overheden en bedrijfsleven bestaat tegenwoordig ook een HandelsRegisterNummer (HRN). Dit OIN of HRN wordt opgenomen in de client en server PKI-Overheid certificaten die worden gebruikt voor de authenticatie van zender en ontvanger. Indien StUF-berichten worden gebonden aan een OSBDigikoppeling profiel, dan dient het element <StUF:organisatie> binnen <StUF:zender> en <StUF:ontvanger> in de stuurgegevens gevuld te worden met het OIN of HRN van de zendende respectievelijk de ontvangende organisatie. Voor nadere informatie over het OIN en HRN wordt verwezen naar de website van OSBde-Logius onder het product Digikoppeling.

Omdat het werken met het [OSBDigikoppeling](#) ebMS profiel en met de door ~~de~~ [OSBDigikoppeling](#) voorgeschreven authenticatie en beveiliging (server én client PKI-certificaten) tamelijk complex is voor ~~kleine~~ organisaties, ~~is-gebruiken zij intern vaak eenvoudiger protocollen en Digikoppeling alleen voor communicatie met (externe) andere overheden.~~ De markt biedt diverse zogenaamde Digikoppeling-adapters gateway is een softwarecomponent OSB gateway ontwikkeld. De OSBde die een [OSBDigikoppeling](#) ebMS en [OSBDigikoppeling](#) WUS profiel vertalen~~en~~ naar eenvoudiger koppelvlakken gebaseerd op [bijvoorbeeld](#) WUS webservices of op JMS_-Gateway JMS profiel. OSB Gateway WUS-lite profiel en het OSB, het ~~zogenaamde~~. Daarnaast zorgt ~~de een~~ [OSBDigikoppeling-adapter-gateway](#) voor het beheren van het certificaat en de afhandeling van de authenticatie en de beveiliging.

De volgende paragrafen gaan in op de binding van StUF-berichten aan [OSBDigikoppeling](#) profielen. Paragraaf 5.1 gaat in op de binding van StUF-berichten aan het [OSBDigikoppeling](#) WUS profiel. Paragraaf 5.2 gaat in op de binding van StUF-berichten aan het [OSBDigikoppeling](#) ebMS profiel. Paragraaf 5.3 gaat in op de binding aan het ~~OSB~~WUS-lite profiel voor ~~de een dergelijke~~ [OSBDigikoppeling-adapter-gateway](#). Vooralnog is geen binding aan het ~~OSB~~JMS-profiel ~~voor de gateway~~ voorzien.

5.1 ~~De b~~Binding van StUF-berichten aan het [OSBDigikoppeling](#) WUS profiel

De eisen voor een [OSBDigikoppeling](#) WUS koppelvlak zijn beschreven in het document '[OSBDigikoppeling](#) Koppelvlakstandaard WUS 1.1'. Dit document bevat diverse profielen die een samenhangende set van functionaliteit bieden. In de loop van de tijd is het aantal profielen waaruit bij implementatie gekozen kan worden uitgebreid; er zijn geen profielen verdwenen. Dit betekent dat oude implementaties altijd voldoen aan de nieuwste versie van de standaard. Tekstuele verbeteringen en bug-fixes van profielen worden alleen aangebracht op het document met de laatste versie. Het is daarom raadzaam om ook bij implementaties die gebaseerd zijn op een profiel van de oude versie altijd de beschrijving van dit profiel in de nieuwste versie te raadplegen. Er is ~~daarnaast~~ het document met de Koppelvlakspecificatie een best practices document [OSBDigikoppeling](#) WUS Best Practices 1.1.

5.1.1 ~~Digikoppeling Koppelvlakstandaard~~ WUS 1.1

Deze paragraaf beschrijft de binding om StUF berichten te kunnen uitwisselen via een [OSBDigikoppeling](#) WUS koppelvlak. Aan de eisen voor een [OSBDigikoppeling](#) WUS koppelvlak beschreven in het document '[OSBDigikoppeling](#) Koppelvlakstandaard WUS 1.1' dient sowieso voldaan te worden. Tevens gelden de eisen die hoofdstuk 4, "StUF-bBinding op basis van WSDL, SOAP en http" stelt aan de definitie van het koppelvlak en de eis voor het vullen van de zendende en de ontvangende organisatie in de stuurgegevens met een OverheidsIdentificatieNummer (~~of~~ OIN) of voor niet-overheidsorganisatie het handelsregisternummer (HRN).

De [OSBDigikoppeling](#) WUS koppelvlakstandaard schrijft het gebruik van een WS-Addressing SOAP-header voor. De onderstaande tabel geeft aan hoe deze WS-Addressing header gevuld wordt op basis van de StUF-stuurgegevens.

WS-Addressing elementen	Vulling op basis van de StUF-stuurgegevens
wsa:To	wsa:To dient gevuld te worden conform de OSBDigikoppeling voorschriften.
wsa:Action	wsa:Action dient in een verzoekbericht gevuld te worden met de namespace van het sectormodel waartoe het bericht behoord gevolgd door een '/' en de waarde van het name attribute van de operation. In de respons wordt als wsa:action gebruikt de wsa:action in het verzoek met daaraan toegevoegd 'Respons'. Als de respons een SOAP fault is, dan wordt als wsa:action gebruikt de wsa:action in het verzoek met daaraan toegevoegd 'Fout'. Voor een verzoekbericht is de waarde van wsa:Action gelijk aan de waarde van het attribute SOAPaction, zoals voorgeschreven in de OSBDigikoppeling koppelvlak specificatie. De waarde voor wsa:Action kan binnen het

WS-Addressing elementen	Vulling op basis van de StUF-stuurgegevens
	<p><portType> element worden vastgelegd door het opnemen van een attribute wsam:Action in het <input>, <output> of <fault> element in een <operation>:</p> <pre> <portType name="BeantwoordVraag"> ... <operation name="npsLv01"> <input message="BG:npsLv01" wsam:action="http://www.egem.nl/StUF/sector/bg/0310/npsLv01"/> <output message="BG:npsLa01" wsam:action="http://www.egem.nl/StUF/sector/bg/0310/npsLv01Respons"/> <fault name="fout" message="StUF:Fo02" wsam:action="http://www.egem.nl/StUF/sector/bg/0310/npsLv01Fout"/> </operation> ... </portType> </pre> <p>met wsam de prefix voor de namespace "http://www.w3.org/2007/05/addressing/metadata/".</p>
wsa:MessageId	wsa:MessageId wordt gevuld met het referentienummer uit de stuurgegevens met daaraan bij voorkeur ² toegevoegd een '@' en een URI die zender van het bericht identificeert. Hiermee wordt alleen voldaan aan de voorkeursvorm UUID@URI, als het referentienummer een UUID is. Bij gebruik van StUF-berichten over de OSBDigikoppeling wordt het gebruik van een UUID als referentienummer aanbevolen.
wsa:RelatesTo	wsa:RelatesTo wordt in de synchrone respons op een ingekomen verzoek gevuld met het crossreferentienummer uit de stuurgegevens met daaraan bij voorkeur toegevoegd een '@' en een URI die zender van het ingekomen verzoek identificeert. Bij synchrone communicatie wordt wsa:RelatesTo zo automatisch gevuld met wsa:MessageId van het verzoek.
Overige elementen	De overige elementen gedefinieerd door de WS-Addressing standaard dienen worden gevuld conform de voorschriften in de OSBDigikoppeling Koppelvlakstandaard WUS 1.1.

Tabel 3: Het vullen van WS-addressing elementen op basis van de stuurgegevens

5.1.2 [Digikoppeling Koppelvlakstandaard WUS 2.0](#)

De binding van StUF berichten aan het Digikoppeling WUS koppelvlak is voor versie 2.0 identiek aan versie 1.1. Voor de overige elementen wordt echter verwezen naar het document Digikoppeling Koppelvlakstandaard WUS 2.0'. Dit geldt specifiek indien profielen voor bijvoorbeeld signing/encryptie op berichtniveau en/of attachments toegepast worden die nog niet voorkwamen in eerdere versies. In geval van binaire bijlagen dient tevens aan de voorschriften in sectie 3.1 te worden voldaan.

5.2 ~~De b~~Binding van StUF-berichten aan het [OSBDigikoppeling](#) ebMS profiel

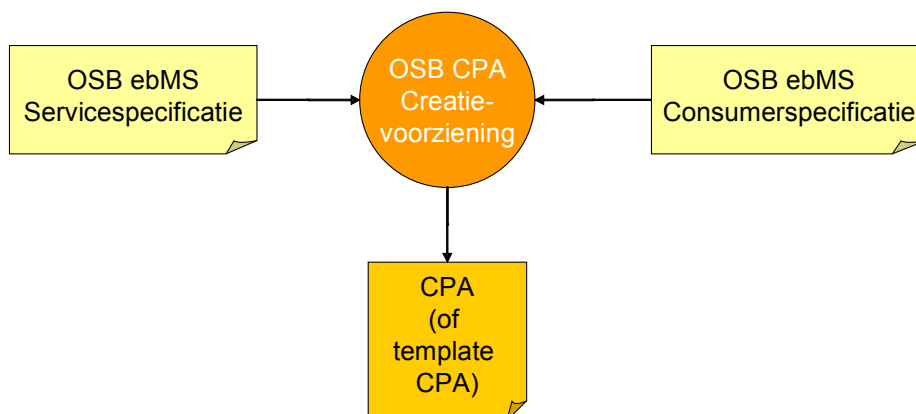
De eisen voor een [OSBDigikoppeling](#) ebMS koppelvlak zijn beschreven in het document 'OSBDigikoppeling Koppelvlakstandaard ebMS-1.1'. Dit document bevat diverse profielen die een samenhangende set van functionaliteit bieden. In de loop van de tijd is het aantal profielen waaruit bij implementatie gekozen kan worden uitgebreid; er zijn geen profielen verdwenen. Dit betekent dat oude implementaties altijd voldoen aan de nieuwste versie van de standaard. Tekstuele verbeteringen en bug-fixes van profielen worden alleen aangebracht op het document met de laatste versie. Het is daarom raadzaam om ook bij implementaties die gebaseerd zijn op een profiel van de oude versie altijd de beschrijving van dit

² Er staat hier bij voorkeur, omdat niet alle SOAP handlers het gebruikt van de [a@b](#) ondersteunen.

profiel in de nieuwste versie te raadplegen. Er is ~~daarnaast~~ het document met de Koppelvlakspecificatie een 'best practices' document- (OSBDigikoppeling ebMS Best Practices 1.1) dat aangeeft hoe omgegaan moet worden met de ebMS specificatie binnen de OSBDigikoppeling. ~~Gateway deze versie op het moment van publicatie nog niet, omdat het in versie 1.1 mogelijk is om de aan een verzoek geassocieerde responsberichten binnen één service te definiëren in plaats van in twee aparte services.~~ OSB ~~Er is aangesloten bij versie 1.1 ook al ondersteunt de~~

Voor de configuratie van een OSBDigikoppeling ebMS koppelvlak in de ebMS adapter van zowel de service requester als de service provider is een aantal gegevens nodig. Deze worden vastgelegd in een CPA – een zogenaamd Collaboration Protocol Agreement (dit is vergelijkbaar met een WSDL voor WUS). Per combinatie van service requester en service provider dient er een CPA gemaakt te worden. Voor één service aangeboden door een service provider zijn er dus net zoveel CPA's als er requesters van deze service zijn.

Omdat er voor één service vaak meerdere CPA's zijn en omdat het maken van CPA's complex is, heeft de OSBDigikoppeling een hulpmiddel ontwikkeld om te komen tot een CPA, de zogenaamde OSBDigikoppeling CPA creatievoorziening. Ten behoeve hiervan wordt voor het OSBDigikoppeling ebMS koppelvlak een service vastgelegd in een door de OSBDigikoppeling gedefinieerd xml-document, de zogenaamde OSBDigikoppeling ebMS Servicespecificatie. Samen met een OSBDigikoppeling ebMS Consumerspecificatie wordt een CPA of template CPA -afgeleid. (Een template CPA kan -worden gebruikt voor het definiëren van CPA als men geen gebruik wil of kan maken van de OSBDigikoppeling CPA Creatievoorziening).



De handleiding hiervan is te vinden in 'OSBDigikoppeling CPA Creatie Handleiding 1.13'. Het Stelselhandboek (www.stelselhandboek.nl) geeft een heldere procesbeschrijving van activiteiten die nodig zijn bij het toepassen van Digikoppeling en CPA's. ~~CPA Creatie Proeesbeschrijving versie 1.1' geeft een eenvoudige procesbeschrijving.~~ OSB ~~Het document'~~

Voor het definiëren van een StUF-koppelvlak met behulp van een wsdl is hierboven een aantal voorschriften gegeven. Hetzelfde doen we voor het definiëren van een OSBDigikoppeling ebMS koppelvlak voor StUF-berichten door enkele voorschriften te geven voor de OSBDigikoppeling ebMS Servicespecificatie.

5.2.1 Digikoppeling Koppelvlakstandaard ebMS 1.1

Het specificeren van een Digikoppeling ebMS koppelvlak voor StUF-berichten

Het specificeren van een OSB ebMS koppelvlak voor StUF-berichten

In paragraaf 4.2 is een beperkt aantal portTypes onderkend en deze portTypes zijn het uitgangspunt voor het definiëren van de onderkende services voor StUF-berichten. Berichtuitwisseling over het OSBDigikoppeling ebMS koppelvlak is altijd asynchroon. Alle asynchrone StUF-berichten worden gedefinieerd als <message> elementen binnen een service waarvan de naam eindigt op OntvangAsynchroon. Bij de binding aan OSBDigikoppeling ebMS worden geen bevestigingsberichten gebruikt om de ontvangst van het bericht te bevestigen naar de verzender. Er kan wel een Fo03-foutbericht worden teruggezonden. Er staat 'in een

service waarvan naam eindigt op', omdat de [OSBDigikoppeling](#) voorschriften geeft om te komen tot een unieke servicenaam. Deze voorschriften dienen uiteraard gevolgd te worden. Als aanscherping op EB006 uit [OSBDigikoppeling](#) Best Practices ebMS schrijft de StUF-binding voor dat de naam van de action gelijk is aan de elementnaam in het <message> element.

Hieronder staat een voorbeeld voor het <messages> element in een [OSBDigikoppeling](#) ebMS Servicespecificatie:

```
<messages process="DeProcesNaam" service="OSBDK Voorloop.OntvangAsynchroon"
  xmlns:xsdns1="http://www.egem.nl/StUF/sector/bg/0310"
  xmlns:xsdns2="http://www.egem.nl/StUF/0301">
  <message from="SR" to="SP" profile="esbosbDK-rm" element="xsdns1:npsLk01">npsLk01</message>
  <message from="SR" to="SP" profile="esbosbDK-rm" element="xsdns1:nraLk01">nraLk01</message>
  ...
  <message from="SP" to="SR" profile="esbosbDK-rm"
    element="xsdns2:Fo03Bericht">Fo03Bericht</message>
</messages>
```

De parameter PartyId is ook relevant bij de binding van StUF-berichten aan het [OSBDigikoppeling](#) ebMS profiel. De waarde van het element <organisatie> binnen <zender> en <ontvanger> in de stuurgegevens dient gelijk te zijn aan het PartyId (het OIN [of HRN](#)) van de Service Requester respectievelijk de Service Provider voor het inkomende verzoekbericht en precies andersom voor het bericht dat als respons wordt gestuurd. Er is een uitzondering op deze regel. Indien conform EB004 uit [OSBDigikoppeling](#) Best Practices ebMS het PartyId is voorzien van een postfix, dan dient het element <administratie> binnen <zender> of <ontvanger> in de stuurgegevens deze postfix met weglating van de '_' als waarde te hebben en dient het element <organisatie> binnen <zender> of <ontvanger> in de stuurgegevens als waarde de OIN zonder postfix te hebben. Een eventueel in de CPA opgenomen PartyName wordt niet vertaald naar een element in de StUF stuurgegevens.

Het voorschrijven van een messageOrder heeft alleen zin voor asynchrone antwoordberichten, omdat dit de enige berichten zijn waarvoor het ConversationId wordt gevuld. Voor het verder vullen van de [OSBDigikoppeling](#) ebMS Servicespecificatie wordt verwezen naar de [OSBDigikoppeling](#) documentatie.

5.2.1.1 ~~Het vullen van de ebMS SOAP header voor StUF-berichten~~

Elke ebMS adapter heeft een eigen leveranciersspecifieke interface voor het aanbieden en ontvangen van berichten door een applicatie. Via dit interface dienen de gegevens in de ebMS SOAP header geleverd cq opgehaald te worden. Een en ander is natuurlijk slechts nodig voorzover de adapter ze niet afleid uit de CPA cq voorzover ze niet bekend zijn in de applicatie. Het interface op zich zal hier niet besproken worden. Wel wordt in onderstaande tabel ingegaan op de waarden voor een aantal van de ebMS header elementen die afgeleid moeten worden uit de CPA of uit de inhoud van het bericht. Voor de overige gegevens, bijvoorbeeld nodig voor de betrouwbare overdracht van berichten, authenticatie, encryptie en dergelijke, wordt verwezen naar de '[OSBDigikoppeling](#) Best Practices ebMS 4.12.0' en de overige voorschriften van de [OSBDigikoppeling](#).

OSBDigikoppeling ebMS header elementen	Vulling op basis van de StUF-stuurgegevens
Service	<eb:Service> wordt gevuld met de naam van de service in de CPA.
Action	<eb:Action> wordt gevuld met de elementnaam van het verzonden bericht.
From	Binnen <eb:From> wordt <eb:PartyId> gevuld met het PartyId in de CPA voor de verzender van het bericht en <eb:Role> wordt gevuld met "SR", als het gaat om een verzoekbericht dat wordt verzonden door de Service Requester en met "SP", als het gaat om een responsbericht dat wordt verzonden door de Service Provider.

OSBDigikoppeling ebMS header elementen	Vulling op basis van de StUF-stuurgegevens
To	Binnen <eb:To> wordt <eb:PartyId> gevuld met het PartyId in de CPA voor de ontvanger van het bericht en <eb:Role> wordt gevuld met “SP”, als het gaat om een verzoekbericht dat wordt verzonden naar de Service Provider en met “SR”, als het gaat om een responsbericht dat wordt verzonden naar de Service Requester.
ConversationId	<eb:ConversationId> wordt alleen gevuld voor asynchrone vraag- en antwoordberichten, omdat één asynchrone vraag met meerdere asynchrone antwoordberichten beantwoord kan worden. De waarde voor <eb:ConversationId> in het vraagbericht is de inhoud van het element <StUF:refNummer> in de stuurgegevens van het vraagbericht en is de inhoud van het element <StUF:crossRefNummer> in de stuurgegevens van het antwoordbericht.
MessageId	<eb:MessageId> wordt gevuld met de inhoud van het element <StUF:referentienummer> in de stuurgegevens geconcateneerd met het “@” teken en de domeinnaam van afzender. Bij gebruik van StUF-berichten over de OSBDigikoppeling wordt het gebruik van een UUID als referentienummer aanbevolen.
RefToMessageId	<eb:RefToMessageId> wordt niet opgenomen, als het element <StUF:crossRefNummer> niet voorkomt in de stuurgegevens. Het wordt gevuld met de inhoud van het element <StUF:crossRefNummer> in de stuurgegevens geconcateneerd met het “@” teken en de domeinnaam van de ontvanger van het bericht, zijnde de zender van het verzoekbericht waar dit bericht een respons op is. (De RefToMessageId in de response is dus gelijk aan het MessageId van het oorspronkelijke request.)

5.2.2 Digikoppeling Koppelvlakstandaard ebMS 2.0

De binding van StUF berichten aan het Digikoppeling ebMS koppelvlak is voor versie 2.0 identiek aan versie 1.1. Voor de specifieke Digikoppeling velden wordt echter verwezen naar het document Digikoppeling Koppelvlakstandaard ebMS 2.0'. Dit geldt specifiek indien profielen voor bijvoorbeeld signing/encryptie op berichtniveau en/of attachments toegepast worden die nog niet voorkwamen in eerdere versies. In geval van binaire bijlagen dient tevens aan de voorschriften in sectie 3.1 te worden voldaan.

5.3 De binding aan het OSB WUS-lite profiel

De binding van StUF berichten aan transport protocollen van en naar digikoppeling software, dat wil zeggen het transport van berichten binnen het organisatie domein heeft een aantal aandachtspunten. Voor gebruik binnen organisaties zijn in hoofdstuk 4 reeds bindingen gedefinieerd.

Voor berichten die in het Digikoppeling domein Synchroon (WUS) zijn moeten intern de bindingen uit hoofdstuk 4 gevolgd worden. Het WS-Addressing element <wsa:From> is voor StUF-berichten niet relevant, omdat het OIN van de organisatie die het verzoek doet is opgenomen in het element <StUF:organisatie> binnen <StUF:zender> in de stuurgegevens. In Het interne koppelvlak richting Digikoppeling adapter hoeft wsa:from dan ook niet gebruikt te worden zolang het in het digikoppeling domein zelf wel juist wordt ingevuld.

Voor A-synchrone berichten, die in het digikoppeling domein via ebMS lopen kan ook voor de bindingen uit hoofdstuk 4 worden gekozen. Hierbij dient opgemerkt te worden dat ebMS een reliable koppelvlak is en de binding uit hoofdstuk 4 dit op koppelvlak niveau niet is. De reliability zal dus op de applicatie laag

~~gegarandeerd moeten worden. Dit kan door in de applicatie niveau niet alleen een WSDL te gebruiken voor het versturen van data maar ook één in te richten voor het ontvangen van een bevestiging.~~

~~Ontvangen Asynchroon bericht:~~

~~Op asynchrone berichten die het Digikoppeling ebMS profiel gebruiken dient een applicatie in het organisatiedomein bij niet te reageren met een Bv03- of Fo03-bericht, maar met een voor de adapter specifiek bericht dat de ontvangst bevestigt³ indien de ontvanger in staat is om het bericht te persisteren. Dit responsebericht dient ook verstuurd te worden, wanneer volgens de StUF-standaard de respons een Fo03-bericht is. Het Fo03-bericht dient vervolgens als een asynchroon bericht te worden verstuurd naar de aanbieder van het bericht. Wanneer de ontvangende applicatie om wat voor reden dan ook niet in staat is om het bericht te persisteren, dan wordt geen respons gezonden en krijgt de Digikoppeling adapter een http-time-out.~~

•

~~Verzenden Asynchroon bericht:~~

~~Ook bij het aanbieden van een asynchroon bericht aan de Digikoppeling adapter zal door deze niet gereageerd worden met een Bv03- of Fo03-bericht, maar met een eigen responsbericht⁴. Bij ontvangst van dit bericht mag de zender ervan uitgaan dat het bericht zal aankomen bij de ontvanger.~~

De binding van StUF-berichten aan het OSB WUS-lite profiel is identiek aan de binding aan het OSBDigikoppeling WUS profiel. Uiteraard dient voldaan te worden aan alle eisen gesteld door het OSB WUS-lite profiel. Het WS-Addressing element <wsa:From> is voor StUF-berichten niet relevant, omdat het OIN van de organisatie die het verzoek doet is opgenomen in het element <StUF:organisatie> binnen <StUF:zender> in de stuurgegevens.

Voor synchrone berichten die op de OSB het Digikoppeling WUS profiel gebruiken, worden bij binding aan het WUS-lite profiel de StUF-voorschriften beschreven in hoofdstuk 4 gevolgd.

Op asynchrone berichten die op de OSB het Digikoppeling ebMS profiel gebruiken dient een applicatie in het organisatiedomein bij gebruik van het WUS-lite profiel niet te reageren met een Bv03- of Fo03-bericht, maar met het hiervoor door het WUS-lite profiel voorgeschreven response bericht gedefinieerd in bijlage A van [OSBGatW]dit document, indien de ontvanger in staat is om het bericht te persisteren. Dit voorgeschreven responsebericht dient ook verstuurd te worden, wanneer volgens de StUF-standaard de respons een Fo03-bericht is. Dit Fo03-bericht dient vervolgens als een asynchroon bericht te worden verstuurd naar de aanbieder van het bericht. Wanneer de ontvangende applicatie om wat voor reden dan ook niet in staat is om het bericht te persisteren, dan wordt geen respons gezonden en krijgt de gatewayDigikoppeling adapter een http-time-out.

Ook bij het aanbieden van een asynchroon bericht aan de OSBDigikoppeling adaptergateway zal niet gereageerd worden met een Bv03- of Fo03-bericht, maar met het door de OSB GatewayDigikoppeling in de bijlage gedefinieerde responsbericht. Bij ontvangst van dit bericht mag de zender ervan uitgaan dat het bericht zal aankomen bij de ontvanger.

³Een voorbeeld hiervan wordt in bijlage A van dit document gegeven

⁴Een voorbeeld hiervan wordt in bijlage A van dit document gegeven

6 Referenties

[HTTP] RFC 2616 - Hypertext Transfer Protocol HTTP/1.1

<http://www.w3.org/Protocols/rfc2616/rfc2616.html>

[MTOM] specificatie van MTOM in combinatie met SOAP1.2

<http://www.w3.org/TR/soap12-mtom/>

[MTOM, SOAP1.1] specificatie van het gebruik van MTOM in combinatie met SOAP1.1

<http://www.w3.org/Submission/soap11mtom10>

[SOAP] Versie 1.1

<http://www.w3.org/TR/2000/NOTE-SOAP-20000508>

[StUF] Versie 0301

[https://www.surfgroepen.nl/sites/stuf/Shared Documents/Forms/bestanden.aspx?](https://www.surfgroepen.nl/sites/stuf/Shared Documents/Forms/bestanden.aspx?RootFolder=/sites/stuf/Shared Documents/4_StUF_Standaarden/StUF 03.01 (In Gebruik))

[RootFolder=/sites/stuf/Shared Documents/4_StUF_Standaarden/StUF 03.01 \(In Gebruik\)](https://www.surfgroepen.nl/sites/stuf/Shared Documents/Forms/bestanden.aspx?RootFolder=/sites/stuf/Shared Documents/4_StUF_Standaarden/StUF 03.01 (In Gebruik))

[StUFXSD] Versie 0301

[https://www.surfgroepen.nl/sites/stuf/Shared Documents/Forms/bestanden.aspx?](https://www.surfgroepen.nl/sites/stuf/Shared Documents/Forms/bestanden.aspx?RootFolder=/sites/stuf/Shared Documents/4_StUF_Standaarden/StUF 03.01 (In Gebruik))

[RootFolder=/sites/stuf/Shared Documents/4_StUF_Standaarden/StUF 03.01 \(In Gebruik\)](https://www.surfgroepen.nl/sites/stuf/Shared Documents/Forms/bestanden.aspx?RootFolder=/sites/stuf/Shared Documents/4_StUF_Standaarden/StUF 03.01 (In Gebruik))

[WSDL] Versie 1.1

<http://www.w3.org/TR/wsdl>

[WSIBP] Versie 1.1

www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html

[xmime] Specificatie van het specificeren van MIME mediatype in berichten en schema's

<http://www.w3.org/TR/xml-media-types/>

[XML] Extensible Markup Language (XML) 1.0 (Second Edition)

<http://www.w3.org/TR/2000/REC-xml-20001006>

[XML Schema]

<http://www.w3.org/TR/2004/REC-xmlschema-0-20041028> (Primer)

<http://www.w3.org/TR/2004/REC-xmlschema-1-20041028> (Structures)

<http://www.w3.org/TR/2004/REC-xmlschema-2-20041028> (Datatypes)

[xop] specificatie van xop gebruikt door MTOM

<http://www.w3.org/TR/xop10/>

[EBMS] ISO 15000-2 ebXML Message Service Specification.

<http://www.oasis-open.org/specs/index.php#ebxmlmsgv2>

[~~OSBDK~~WUS] Versie ~~2.01~~+

http://www.logius.nl/fileadmin/logius/product/digikoppeling/koppelvlakstandaarden/Koppelvlakstandaard%20WUS_Digikoppeling%20v2.2.pdf

~~/WUS_1.1_KP.pdf~~~~OSB~~<http://www.overheidsservicebus.nl/fileadmin/>

[~~OSBDK~~WUSBP] Versie ~~2.0~~+

~~—~~<http://www.logius.nl/fileadmin/logius/product/digikoppeling/koppelvlakstandaarden/Digikoppeling%20Best%20Practices%20WUS%20v%201.2.pdf>

~~—Best_Practiees_WUS_1_1.pdf~~~~OSB/OSB~~<http://www.overheidsservicebus.nl/fileadmin/>

[~~OSB~~~~DK~~ebMS] Versie ~~1.1~~~~2.0~~

~~http://www.logius.nl/fileadmin/logius/product/digikoppeling/koppelvlakstandaarden/Koppelvlakstandaard%20ebMS_Digikoppeling%20_v2.2.pdf~~
~~[_Koppelvlakstandaard_cbMS_v1_1.pdf](http://www.overheidsservicebus.nl/fileadmin/_Koppelvlakstandaard_cbMS_v1_1.pdf)~~~~OSB/OSB~~~~<http://www.overheidsservicebus.nl/fileadmin/>~~

[~~OSB~~~~DK~~ebMSBP] Versie ~~1.1~~~~2.0~~

~~<http://www.logius.nl/fileadmin/logius/product/digikoppeling/koppelvlakstandaarden/Digikoppeling%20Best%20Practices%20ebMS%20v1.2.pdf>~~
~~[_Best_Practices_cbMS_v1_3.pdf](http://www.overheidsservicebus.nl/fileadmin/_Best_Practices_cbMS_v1_3.pdf)~~~~OSB/OSB~~~~<http://www.overheidsservicebus.nl/fileadmin/>~~

[~~OSB~~GatW] Versie 1.0

~~[_Gateway_-_Functionele_Specificatie_v1.0.pdf](http://www.overheidsservicebus.nl/fileadmin/_Gateway_-_Functionele_Specificatie_v1.0.pdf)~~~~OSB/OSB~~~~<http://www.overheidsservicebus.nl/fileadmin/>~~

Bijlage A – XSD - WUS-lite response

```
<?xml version="1.0" encoding="UTF-8" ?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="stringtype">

    <xs:restriction base="xs:string"/>

  </xs:simpleType>

  <xs:complexType name="responsetype">

    <xs:sequence>

      <xs:element name="text" type="stringtype"/>

    </xs:sequence>

  </xs:complexType>

  <xs:element name="response" type="responsetype"/>

</xs:schema>
```

=

Bijlage B – Koppelvlakstandaard WUS-lite

Inleiding

De inhoud van deze WUS-lite standaard bouwt voort op de Digikoppeling koppelvlakstandaard WUS en de daarin genoemde onderliggende internationale standaarden. In de Digikoppeling koppelvlakstandaard WUS zijn een aantal voorschriften gedefinieerd. In dit hoofdstuk worden wijzigingen ten aanzien van deze voorschriften opgenomen die gelden voor het WUS-lite koppelvlak.

Een web service wordt deels gedefinieerd (beschreven – description) door een WSDL. Deze WSDL geeft een beschrijving van de eisen die ten aanzien van de communicatie gesteld worden. Het is een abstracte definitie van de web service. De web service communiceert feitelijk middels SOAP berichten, die gegenereerd worden op basis van de WSDL. De WUSlite koppelvlakstandaard beschrijft per definitie aan welke eisen een WSDL moet voldoen, die het koppelvlak beschrijft tussen de Digikoppeling adapter en de organisatieapplicaties.

Omdat de koppelvlakstandaard WUS Lite een afgeleide is van de Digikoppeling koppelvlakstandaard WUS, heeft het de voorkeur om alleen de verschillen die gelden met de Digikoppeling koppelvlakstandaard WUS in deze bijlage te beschrijven. Als basis hiervoor geldt dus de Digikoppeling Koppelvlakstandaard WUS.

De verschillen die worden belicht betreft de volgende onderdelen:

-

- TLS

WS-Addressing

-

Geen TLS

De Koppelvlakstandaard WUS-lite definieert communicatie in het Organisatie-Domein. Het voorziet niet in HTTP over TLS/SSL communicatie met de gedachte dat een dergelijke beveiliging in het organisatie-domein niet noodzakelijk is. Dit neemt de complexiteit die TLS/SSL met zich meebrengt weg en biedt zo vereenvoudigde implementatie mogelijkheden.

Wijziging WS-Addressing

Voor de WUS-lite specificatie geldt de volgende wijzigingen ten opzichte van wat is beschreven in de Digikoppeling koppelvlakstandaard WUS.

-

In het het Organisatie-Domein wordt het WS-Addressing From element gebruikt.

-

wsa:From

Het WS-Addressing From element wordt gebruikt voor het opnemen van de identiteit van de Service-Requester.

Een request bericht dat afkomstig is uit het Digikoppeling domein wordt doorgestuurd naar de applicatie in het Organisatie-Domein waarvoor het bedoeld is. De waarde van het veld Subject.serialnumber uit het PKIoverheid certificaat wordt geplaatst in het wsa:From element van de bericht header dat naar de organisatieapplicatie verstuurd wordt. Voor de aanduiding van het certificaat subject.serialnumber wordt het

element “identifier” van de Dublin Core namespace “http://purl.org/dc/elements/1.1/” gebruikt. Hieronder staat een mogelijk voorbeeld van de vulling van het wsa:From element bij een request in het Organisatie-Domein:

```
<wsa:From>
  <wsa:Address> http://www.w3.org/2005/08/addressing/anonymous
</wsa:Address>
  <wsa:ReferenceParameters>
    <de:identifier>
      xmlns:de="http://purl.org/dc/elements/1.1/">xxxxxxxxxxxx
    </de:identifier>
  </wsa:ReferenceParameters>
</wsa:From>
```

-

Het wsa:From element dat in de response naar de Service Requester in het Digikoppeling-Domein gaat wordt weggelaten, of voldoet zoals in Digikoppeling Koppelvlakstandaard voorgeschreven.

-

Aandachtspunten

De WUS-lite standaard kent nog een aantal aandachtspunten. Dit zijn voornamelijk zaken waar voorlopige keuzes zijn gemaakt, maar welke in de toekomst expliciet getoetst moeten worden. Er gelden momenteel de volgende aandachtspunten:

-

- Porttype in WSDL voor WUS-Lite bij ebMS communicatie in het Digikoppeling Domein

Porttype in WSDL voor WUS-Lite bij ebMS communicatie in het Digikoppeling-Domein

De communicatie via ebMS is asynchroon en in essentie wordt als MEP alleen “one-way” ondersteund. Het WUS-lite koppelvlak zou in dit geval dus eigenlijk geen response bericht moeten hebben. Dit kan worden vertaald door bij de contracttransformatie voor het WUS-Lite koppelvlak een zogenaamde “one-way operation” WSDL te genereren. In dit geval zal de Service Requester een SOAP bericht in het Organisatie-Domein versturen naar de Service Provider (de Gateway), waarbij er geen SOAP response bericht terug komt. Er wordt enkel via HTTP een response gegeven (code 200/202 indien op HTTP niveau alles goed ging).

Het nadeel bij deze optie is, is dat er geen (SOAP) fouten teruggestuurd kunnen worden en de ServiceRequester alleen HTTP status codes van het response heeft om te bepalen dat het bericht verstuurd is. Het heeft daarom momenteel de voorkeur om ook bij ebMS communicatie in het Digikoppeling-Domein het WUS-lite koppelvlak te definiëren volgens een request/response operation.

Een bijkomend voordeel hiervan is, is dat het WUS-lite koppelvlak generiek ingericht kan worden voor zowel bevestigingen (Digikoppeling-WUS request/respons) als meldingen (Digikoppeling-ebMS one-way).

Wel heeft de keuze om gewoon altijd een “request-response operation” toe te passen tot gevolg dat de Gateway een standaard response moet definiëren in het geval van een Digikoppeling-Domein ebMS naar WUS-lite contracttransformatie. De Service Requester dient er rekening mee te houden dat het response bericht in dit geval niets zegt over de daadwerkelijke verwerking van het bericht in het Digikoppeling-Domein, maar alleen dat het bericht door de Gateway ontvangen is.